International Journal of scientific research and management (IJSRM)
||Volume||2||Issue||5 ||Pages|| 925-934 ||2014||
Website: www.ijsrm.in ISSN (e): 2321-3418

Application of Discrete Optimization Technique Hybrid JPSO/LIDM to Optimal Design of Pressurized Irrigation Networks

Rasoul Rajabpour [1*], *Bahram Sami Kashkoli* [2]

1-Department of Civil Engineering, Beyza Branch, Islamic Azad University, Beyza
2-Rahab Andishan Zagros Co., Shiraz

Absstract : The layout of irrigation water distribution networks is usually branched. In most of the previous studies pipe sizes of the network had been optimized assuming a predetermined layout of the distribution system. However, only few researches have focused on the simultaneous layout and pipe size optimization. In this study a hybrid approach is adapted for simultaneous layout and pipe size optimization of branched pipe networks as a cost minimization problem. This new approach is based on combination of a pipe size optimizer (LIDM) with a layout optimizer for joint layout and pipe size optimization. At each iteration, the layout optimizer algorithm acts as an outer loop and LIDM acts as an inner loop. Once all the solutions are developed (each solution is a specific layout), LIDM can be used to optimize pipe sizes of each developed branched layout (solution). Then solution costs can be calculated and according to them, layout optimizer rearranges the solutions and the process continues. Two different approaches are used for layout optimization. At the First approach by using the loop model each branched layout is encoded as a string of eliminating links and a Discrete Particle Swarm Optimization for combinatorial optimization, called JPSO is applied to select the best sting of eliminating links. Proposed methods are applied for simultaneous layout and pipe size optimization of a small benchmark example in the literature and the results are presented and compared to the existing results. The results showed that the developed methods have significant advantages compared to other methods used.

Keywords: Distribution Networks; Optimization; Branched Layout; JPSO; Hybrid Approach

1- Introduction

The pressurized systems which were developed during the previous decades had considerable advantages compared to open canals. In fact, they guarantee better services to the users and higher distribution efficiency. They overcome the topographical constraints and make it easier to measure the water volume delivered. Operation, maintenance and management activities are more technical and easier to control (Lamaddalena and Sagardoy, 2000). In order to reduce the costs of the network, the layout of pressurized irrigation water distribution networks are branched. One of the important problems which should be considered in designing such networks is the simultaneous optimization of both the layout and the pipe sizes. The optimization of branched irrigation networks has attracted the attention of some researchers. Some researchers such as Labye (1981) only considered pipe size optimization of branched networks with a predetermined layout. He proposed an approach called Labye's Iterative Discontinuous Method

(LIDM) for optimizing pipe sizes. Using LIDM enables the optimization process to allocate mixed pipe diameters.

Some researchers, on the other hand, have addressed the layout optimization of the branched pipe networks, neglecting the influence of the pipe sizes. Cembrowicz (1992) encoded the problem of branched layout optimization with Loop Model and applied GA for solving it. Walters and Smith (1995) used evolutionary algorithm (EA) for the selection of a branched network from a non-directed base graph. Geem et al. (2000) applied Harmony Search (HS) for the optimal design of branched networks.

Afshar (2006 and 2007) tackled the problem of joint layout and pipe size optimization of branched networks, by considering pipe sizes of maximum layout (a predefined layout that includes all possible links) as decision variables. The pipe sizes can be selected from a set of available pipe sizes. Adding a zero pipe diameter to the list of available pipe diameters would enable the optimization algorithm to remove if required, pipes from the network in its search towards an optimal tree layout. The search space of the optimization model in this formulation consists of all sub-networks of maximum layout. The search space rapidly grows with increasing network size and the number of available pipe sizes. For example the total number of possible combinations for a 3node $*$ 3node base graph of 12 links with 13 possible pipe sizes is 1412. A Large number of these possible combinations are, however, infeasible solutions which, if identified, could be excluded, thus leading to a much smaller search space. In another research Afshar (2005) applied Ant Algorithm (ACO) for simultaneous layout and size optimization of tree-like pipe networks. In this approach ants have to search for the optimal solution in the region which contains networks with feasible layout. The decision points of the problem are associated with the nodes of the maximum layout, and the solution components considered as the list of allowable pipe sizes of links in maximum layout.

In this study a new approach is introduced for simultaneous layout and pipe size optimization of branched pipe networks as a cost minimization problem. This new approach is based on combination of a pipe size optimizer (LIDM) with a layout optimizer. Adapting such hybrid approach to tackle the problem of joint layout and pipe size optimization of branched networks leads to much smaller search space in comparison with ACO application as reported by Afshar (2006). Labye (1981) proposed LIDM for optimizing pipe sizes in a branched irrigation network with a predetermined layout. Using LIDM enables the optimization process to allocate mixed pipe diameters to some links and therefore reduces the costs even more.

In this paper, two different approaches are used for layout optimization. At the First approach by using the loop model each branched layout is encoded as a string of eliminating links and a Discrete Particle Swarm Optimization for combinatorial optimization, called JPSO is applied to select the best sting of eliminating links. To avoid the problems associated with the Loop Model, at the second approach instead of coding the solutions into strings, an especial form of Genetic Algorithm which includes a tree growing algorithm within its "reproduction" phase is used as the layout optimizer.

1-1- Discrete Particle Swarm Optimization

The standard PSO considers a swarm S containing s particles (S = 1, 2… s) in a d-dimensional continuous solution space (Kennedy and Eberhart, 2001). Each i-th particle of the swarm has a position $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ and a velocity $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$. The position X_i represents a solution to the problem, while the velocity V_i gives the rate of change for the position of particle i at the next iteration. The position of particle i in each iteration is adjusted according to:

$$X_i' = X_i + V_i' \quad (1)$$

in which, (′) indicates new values.

Each particle i of the swarm communicates with a social environment or neighborhood, $N(i) \subseteq S$, representing the group of particles with which it communicates, and which could change dynamically. In nature, a bird adjusts its position in order to find a better position, according to its own experience and the experience of its companions. In the same manner, in each iteration, each particle i updates its velocity reflecting the attractiveness of its best position so far $B_i = (b_{i1}, b_{i2}, \dots, b_{id})$ and the best position $G^* = (g_1, g_2, \dots, g_d)$ of its social neighborhood N(i), according to the equation:

$$V_i' = \omega V_i + c_1 \mathrm{rand}(\,)(B_i - X_i) + c_2 \mathrm{rand}(\,)(G^* - X_i) \quad (2)$$

The parameters ω, c_1 and c_2 are positive constant weights representing the degrees of confidence of particle i in the different positions that influence its dynamics, while $\mathrm{rand}(\,)$ refers to a random number with uniform distribution [0, 1] that is independently generated at each iteration.

The original PSO algorithm can only optimize problems in which the elements of the solution are continuous real numbers. Therefore several Discrete Particle Swarm Optimization (DPSO) methods have been proposed. In the DPSO proposed by Kennedy and Eberhart (1997) for problems with binary variables, the position of each particle is a vector $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ of the d-dimensional binary solution space, $X_i \in \{0,1\}^d$, but the velocity is still a vector ,V_i of the d-dimensional continuous space, $V_i \in \mathcal{R}^d$. A different way to update the velocity was considered by Yang et al. (2004).

A DPSO whose particles at each iteration are affected alternatively by its best position and the best position among its neighbors was proposed by Al-kazemi and Mohan (2002). The multi-valued PSO (MVPSO) proposed by Pugh and Martinoli (2006) deals with variables with multiple discrete values. The position of each particle is a mono dimensional array in the case of a continuous PSO, a 2 dimensional array in the case of a DPSO, and a 3-dimensional array for a MVPSO. Indeed, the position of particle i in the MVPSO is expressed by the term x, representing the probability that the ijk i-th particle, in the j-th iteration, takes the k-th value.

A new DPSO proposed in (Moreno-Perez et al., 2007) and (Martinez-Garcia and Moreno-Perez, 2008) does not consider any velocity since, from the lack of continuity of the movement in a discrete space, the notion of velocity loses sense; however they kept the attraction of the best positions. They interpret the weights of the updating equation as probabilities that, at each iteration, each particle has a random behavior, or acts in away guided by the effect of an attraction. The moves in a discrete or combinatorial space are jumps from one solution to another. The attraction causes the given particle to move towards this attractor if it results in an improved solution. An inspiration from the nature for this process is found in frogs, which jump from a lily pad to a pad in a pool. Thus, this new discrete PSO is called Jumping Particle Swarm Optimization (JPSO). Consoli et al. tested capabilities of JPSO by solving minimum labeling Steiner tree problem, an NP-hard graph problem. Based on their computational analysis, JPSO clearly outperformed all the other procedures, obtaining high-quality solutions in short computational running times. This confirms the ability of JPSO method to deal with NP-hard combinatorial problems.

2. Labye's Iterative Discontinuous Method (LIDM)

The approach proposed by Labye (1981), called Labye's Iterative Discontinuous Method (LIDM), for optimizing pipe sizes in an irrigation network is described in this section. This method is developed in two stages.

In the first stage, an initial solution is constructed giving, for each section k of the network, the minimum commercial diameter (D_{min}) according to the maximum allowable flow velocity (v_{max}) in a pipe, when the pipe conveys the calculated discharge (Q_l). After knowing the initial diameters, it is possible to calculate the piezometric elevation $Z_{0,in}$ at the upstream end of the network, which satisfies the minimum head ($H_{j,\,min}$) required at the most unfavorable hydrant j:

$$Z_{0,in} = H_{min} + z_i + h_j \qquad (3)$$

Here h_j is the total head losses along the pathway connecting the hydrant j to the upstream end of the network. Head losses are computed with Darcy-Weisbach equation. The diameters $D_{l,\,min}$ and the initial piezometric elevation $Z_{0,in}$ constitute the initial set of parameters for the optimization of the pipe diameters. This is performed by an iterative procedure.

At each iteration, only two pipes (p and p+1) are considered for each branch. Their diameters, are D_p and D_{p+1} where $D_{p+1} > D_p$. Afterward it is possible to compute the coefficient β as follows:

$$\beta_p = (C_{p+1} - C_p)/(J_p + J_{p+1}) \qquad (4)$$

where C and J are the cost and the friction loss per unit length of pipe, respectively.. Considering any branching sub-network (SN) at the end of the section l, it is possible to minimize the variation of costs ΔC of the network SN^* (Figure 1) using linear programming of equations 5 and 6:

$$\Delta C = -\beta_{p,SN}\Delta Z - \beta_{p,1}\Delta h_1 \qquad (5)$$

Subject to:

$$\Delta Z + \Delta h_1 = \Delta Z'$$

where ΔZ is the variation of the head in the upstream head of (SN), and Δh_l is the variation in the head losses at section l due to the changes of pipe diameters. The optimal solution of equations (5) and (6) produces $\Delta Z = \Delta Z'$ and $\Delta h_l = 0$ when $\beta_{p,\,SN} < \beta_{p,\,l}$, and $\Delta Z = 0$ and $\Delta h_l = \Delta Z'$ when $\beta_{p,\,SN} > \beta_{p,\,l}$. As a result the minimum value for ΔC is

$$\Delta C = -\beta \times \Delta Z$$

where $\beta^* = \min(\beta_{p,SN}, \beta_{p,l})$. The slope of $\beta_{p,SN}$ is

$$\beta_{p,SN} = \beta_{p,l} + \beta_{p,2}$$

when SN is constituted by two sections (1 and 2) in derivation (Figure 1), and

$$\beta_{p,SN} = \min(\beta_{p,l}, \beta_{p,2})$$

when sections 1 and 2 are in series; $\beta_{p,1}$ and $\beta_{p,2}$ are the coefficients defined in equation (4) for the branch 1 and 2 (Figure. 1), with $\beta_{p,l} = 0$ at the terminal sections having excess of head (at the downstream end node).

The procedure is performed starting the optimization from the downstream sections. The magnitude of ΔZ_{iter} for each iteration is:

$$\Delta Z_{iter} = \min[EZ_{iter}, \Delta h_{iter}, (Z_{0,iter} - Z_0)]$$

where EZ_{iter} is the minimum observed for the excess of charge [m] in the nodes where the head changes at each iteration, Δh_{iter} is the minimum value of the head losses variation [m] in those sections where diameters change during the same iteration, and $(Z_{0,iter} - Z_0)$ is the difference between computed and actual piezometric heads [m] at the upstream end. The iterative procedure continues until Z_0 is satisfied.

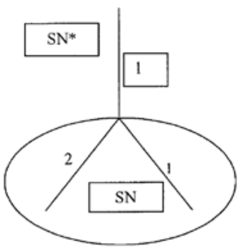

Figure. 1. Elementary network scheme [2].

3- Hybrid Approach

Hybrid approach is based on combination of a pipe size optimizer (LIDM) with a layout optimizer for joint layout and pipe size optimization. In this approach, in each iteration, the layout optimizer algorithm acts as an outer loop and LIDM acts as an inner loop. Once all the solutions are developed (each solution is a specific layout), Labye's Iterative Discontinuous Method can be used to optimize pipe sizes of each developed branched layout (solution). Then solutions costs can be calculated and according to them, layout optimizer rearranges the solutions and the process continues. The basic algorithm of this approach is outlined below:

1- Form an Initial population of a random Tree layouts
2- Optimize the pipe sizes in each layout (pipe size optimization algorithm)
3- Evaluate cost of each layout
4- Generate a new set of tree layouts (using layout optimization algorithm)
5- Repeat from (2), until some termination criterion is reached.

Using LIDM as a pipe size optimizer enables the optimization process to allocate mixed pipe diameters to some links. LIDM chooses optimum diameters for network pipes in such a way following constrains hold satisfied: Hydraulic constrains; Nodal head and pipe flow velocity constrains a Pipe size availability constrains. On the other hand layout optimization algorithm (layout optimizer) selects the least cost spanning tree through a maximum layout, which includes all possible links.

4- Loop Model

Cembrowicz (1992) encoded the problem of branched layout optimization with Loop Model and applied GA for solving it. Cembrowicz (1992) uses graph theory in describing his evolutionary design approach to the problem of branched layout optimization. First he identifies loops within the base graph, and then points out that a tree network can be formed if one link is removed from each loop. The links to be removed (referred to as chords) then become the design variables of the layout problem, which is now one of selecting the set of chords (the co-tree) that, when removed from the base graph, results in the minimum cost spanning tree. He called this approach Loop Model.

The main draw-back of Loop Model is that the removal of an arbitrary link from each loop does not necessarily result in a tree, as illustrated in Figure 3, so a check is necessary on the resulting graph to ensure feasibility. As the network size increases, so does the chance of forming an infeasible network. Moreover water distribution networks with two or more recourses cannot be encoded by Loop Model. Cembrowicz has used the model with success one base graph with 34 nodes, 48 links and 15 loops. Figure 2 shows coding a spanning tree through maximum layout with Loop Model.

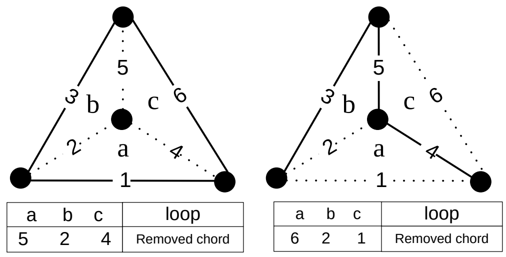

Figure 2 coding a spanning tree through maximum layout with Loop Model

5- Using JPSO for branched layout optimization encoded with Loop Model

The JPSO used for branched layout optimization encoded with Loop Model, considers a swarm S containing s particles ($S = 1, 2 \ldots s$) whose positions $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ evolve in the solution space, jumping from one solution to another. The position of each particle is encoded as a feasible solution to the branched layout optimization problem. At each iteration, there are four different forms of selectable jumps for each particle, from which one should be selected. First one is a random jump (Type one). This kind of jump consists of selecting at random a feature of the solution and changing its value. Second is a jump toward particle`s best position so far $B_i = (b_{i1}, b_{i2}, \dots, b_{id})$ (Type one). Third is a jump toward the best position of particle`s local neighborhood $G_i = (g_{i1}, g_{i2}, \dots, g_{id})$ (Type there) and forth is a jump toward the best particle in the current iteration, which is called global best $G^* = (g_1, g_2, \dots, g_d)$ (Type four). B_i, G_i and G^* are so called attractors. A jump approaching an attractor consists of changing a feature of the current solution by a feature of the attractor. The process of position updating can be described with eq. 11.

$$X_i = c_1 X_i \oplus c_2 B_i \oplus c_3 G_i \oplus c_4 G^* \quad (11)$$

Equation 11 shows that random jump take place with the probability c_1 and jumps toward attractors B_i, G_i and G^* take place with probabilities c_2, c_3 and $c_4 = 1 - (c_1 + c_2 + c_3)$, respectively. At each iteration in order to update the position with this operation, the unit interval [0, 1] is divided into four intervals $[0, c_1)$, $[c_1, c_1+c_2)$, $[c_1+c_2, c_1+c_2+c_3)$ and $[c_1+c_2+c_3, 1]$, which are representatives of jumps type one, two, there and four, respectively. Then a random number is generated with uniform distribution in [0, 1] and based on the interval to which the generated random number belongs, the corresponding type of jump would be selected. As the factors c_1, c_2, c_3 and c_4 control the balance between global and local search, we suggest to have c_1 decrease and c_2, c_3 and c_4 increase linearly with time, in a way to first emphasize global search and then, with each cycle of the iteration, to prioritize local search.

After selecting jump type i with the probability c_i, we choose one of the decision variables randomly with uniform probability. If the selected jump was type one then we change the value of chosen decision variable randomly, else, substitute value of chosen decision variable with corresponding one in the attractor. At the second step we choose a random number ζ with uniform distribution [0, 1]. If $\zeta < c_i$ then we choose another decision variable randomly and change its value again and repeat this step; otherwise the jump stops. Figure 3 illustrates Jump toward an attractor for particles encoded with Loop Model.

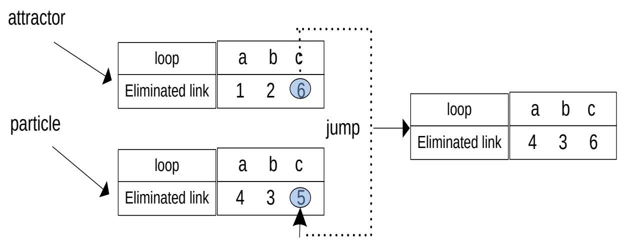

Figure 3 Jump toward an attractor (Particles encoded with Loop Model)

JPSO algorithm is described below:

Note: Each branched layout is encoded as a string of eliminating links.

1. Initial generation of a random population of **m** particles
2. Objective function evaluation
   - 2.1. Set i =1
   - 2.2. If i>m then go to 3
   - 2.3. If the i-th particle was infeasible then set its cost equal to **P** (penalty cost) and go to 2.5 else go to 2.4
   - 2.4. Using LIDM determine optimal diameters for links of i-th particle and calculate its cost
   - 2.5. Set i=i+1 and go to 2-2
3. Update B_i , G_i and G^*
4. Update c_1, c_2 , c_3 and c_4 (equation 11) (Note: during the iterations c_1 linearly decreases and
5. Update particle positions
   - 5.1. Set i=1
   - 5.2. If i>m then go to 6
   - 5.3. Select an attractor for i-th particle
     Generate random number R (0<R<1)
     If $0<R<c_1$ then jumps take place randomly and ($c_i \leftarrow c_1$)
     If $c_1<R<c_1+c_2$ then (attractor$\leftarrow B_i$) and ($c_i \leftarrow c_2$)

If $c_1+c_2<R<c_1+c_2+c_3$ then (attractor←G_i) and (c_i←c_3)
If $c_1+c_2+c_3<R<1$ then (attractor←G^*) and (c_i←c_4)

5.4. Perform jumps in i-th particle

5.4.1. Generate random number ζ ($0 \le \zeta \le 1$)

5.4.2. If $\zeta<c_i$ then perform jump and return to 5.4.1, else go to 5.5

5.5. Set i=i+1 and go to 5.2

6. If termination criteria was met, then algorithm stops else return to 2

JPSO Parameters

Swarm size (s), factors c_1, c_2, c_3 and c_4, maximum number of iterations (m) and neighborhood status should be determined as input parameters. Neighborhood status considered to be circular with radius one. So that each particle has one right neighbor and one left neighbor. Factors c_1, c_2, c_3 and c_4 considered to vary linearly during the iterations. The value of c_1 starts from an initial value, c_1^i and decreases constantly so that at the last iteration (m-th iteration) reaches to its final value c_1^f. The rate of decrease in c_1, Δc, is equal to $(c_1^i - c_1^f)/m$. The initial values of Factors c_2, c_3 and c_4 (c_2^i, c_3^i and c_4^i) increase with a constant rate of $\Delta c/3$ during the iterations.

7- Model application

The proposed methods are applied for simultaneous layout and pipe size optimization of two benchmark examples in the literature.

The example to be considered is that of a simple network shown if Figure 4(a). The network consists of nine nodes, twelve links, and a source located at node number 9. This example has been considered as a test network by Geem et al. (2000) to test the performance of the harmony search method proposed for layout optimization. Afshar (2005, 2006 and 2007) used this example to emphasize the necessity of joint layout pipe size optimization. The water demand at each node of the network is shown in Table 1. Table 2 shows costs of different pipe sizes. The Hazen-Williams coefficient is assumed equal to 130 for all the pipes. The elevation of all the demand nodes is set equal to zero and that of the source node is assumed to be 50 m. Minimum pressure requirement of 30 meter is used at all the demand nodes.

Table 1. Nodal Demand for Network

Node	1	2	3	4	5	5	7	8	9
Demand (l/s)	10	20	10	20	10	20	10	20	…

TABLE 2. Cost Data for Network

Diameter(mm)	1	2	3	4	6	8	10	12	14	16	18	20	22
Cost($/m)	2	5	8	11	16	23	32	50	60	90	130	170	300

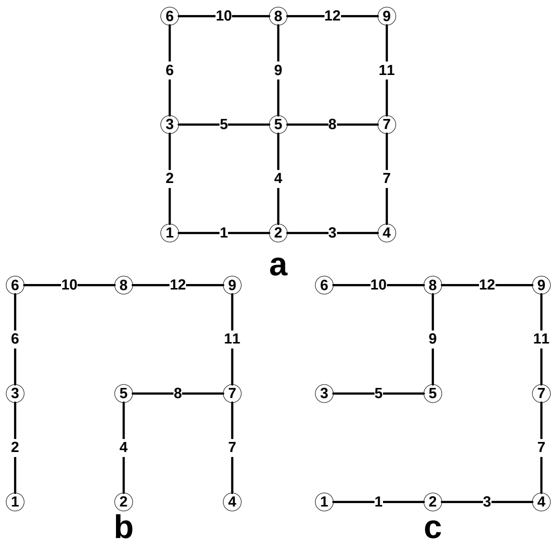

Figure 4 a) Maximum layout of Network1, b) Optimal Layout Obtained for Network 1, Afshar (2005), c) Optimal Layout Obtained for Network 1 with Proposed Methods

Figure 4(c) and table 3 show the resulting layout, pipe diameters and nodal heads obtained using the following values for JPSO parameters: s= 10, m=30, c_1^i =0.55, c_1^f=0.2, c_2^i= 0.15, c_3^i =0.15 and c_4^i=0.15. It should be remarked that the method was able to find the solution of $37,764 within 160 network evaluations. On the other hand, the Genetic Algorithm was able to find a solution same as the solution found by JPSO faster and within only 30 network evaluations. This solution is obtained considering Population Size= 10, Elite count=1 and Crossover fraction: 0.8.

TABLE 3. Optimal Layout and Pipe Sizes for Network

Link	D1(cm)	L1(m)	D2(cm)	L2(m)	Node	P(m)
1	10	100	…	…	1	30.02
2	…	…	…	…	2	31.93
3	14	100	…	…	3	30.62
4	…	…	…	…	4	34.76
5	8	26.13	6	73.87	5	32.9
6	…	…	…	…	6	30.02
7	14	100	…	…	7	42.05
8	…	…	…	…	8	39.78
9	10	100	…	…	9	…
10	8	21.34	6	78.66		
11	14	53.62	12	46.38		
12	14	100	…	…		
Cost		37764.1				

8- Discussion and conclusion

Tables 4 show using hybrid approach leads to better solutions in too much smaller numbers of function evaluations for both of considered example. Here it should be noted that CPU time for each function evaluation at non hybrid approaches (Afshar, 2005, 2006 and 2007) is smaller than corresponding one at hybrid approach. Because, function evaluation at non hybrid approaches involves network hydraulic analysis, but calculating objective function at hybrid approach consists of applying LIDM to optimize pipe sizes which takes more time. In example 1, it should be mentioned that greater performance of Hybrid approach is not only because of LIDM's abilities in pipe sizing but also the obtained layout is better. To prove this, we optimized pipe sizes of layout shown in figure 4(b) (Layout of best solution among non hybrid methods) by LIDM which resulted in solution of $38,301which is still more expensive in compare to $37,761 (Cost of solution obtained by all of hybrid methods for example).

According to Trent (1954), the total number of spanning trees that can be formed from maximum layout of example 1 is 192 which is search space size of hybrid approach. As shown in table 5, adapting hybrid approach to tackle the problem of joint layout and pipe size optimization leads to much smaller search space for layout optimizer in comparison with other approaches (Afshar, 2005, 2006 and 2007). Greater performance of hybrid methods is mainly due to smaller size of search space and strong pipe sizing algorithm.

TABLE 4. Summary of results obtained by different researchers for Network

Method	Cost of best soluti	No. of functi on evalua

		on ($)	tions
1	Separate Layout-Size Optimization (Afshar , 2007)	39,800	…
2	GA (Afshar , 2007)	39,400	7,500
3	ACO (Afshar , 2006)	39,500	8,900
4	ACO (Afshar , 2005)	38,600	3,500
5	Hybrid ACO/LIDM (Kashkooli and Monem, 2009)	37,761	20
6	JPSO	37,761	160
7	Genetic Algorithm	37,761	30

TABLE 5. Search space dimensions considered at different approaches for simultaneous layout and pipe size optimization of example

Method	Search space dimensions	Relative dimensions
GA (Afshar , 2007) and ACO (Afshar , 2006)	$(13+1)^{12}$	$2.95*10^{11}$
ACO (Afshar , 2005)	$192*13^8$	13^8
Hybrid methods	192	1

References

1. Afshar, M.H., Layout and Size Optimization of TREE-LIKE Pipe Networks by Incrementally Solution Building Ants, Proceeding of the First International Conference on Modeling Simulation and Applied Optimization, Sharjah, U.A.E. February 1-3, 2005.
2. Afshar, M.H., Application of a max–min ant system to joint layout and size optimization of pipe networks, 2006, Engineering Optimization, Vol. 38, No. 3, 299–317.
3. Afshar, M.H., Evaluation of Selection Algorithms for Simultaneous Layout and Pipe Size Optimization of Water Distribution Networks, Scientia Iranica, 2007, Vol. 14, No. 1, pp 23-32.
4. Al-kazemi, B. and Mohan, C.K. (2002), Multi-phase Discrete Particle Swarm Optimization. In: Fourth International Workshop on Frontiers in Evolutionary Algorithms, Kinsale, Ireland.
5. Cembrowciz, R.G., Water supply systems optimisation for developing countries. In Pipeline Systems, edited by B. Coulbeck and E. Evans, pp. 59–76, 1992 (Kluwer Academic: London).
6. Consoli, S., Moreno-Perez, J.A., Darby-Downman, k. and Mladenvic, N., Discrete Particle Swarm Optimization for the minimum labeling Steiner tree problem, Studies in Computational Intelligence (SCI), 2008, Vol. 129, 313-322.
7. Martinez-Garcia F.J., Moreno-Perez, J.A. (2008) Jumping frogs optimization: a new swarm method for discrete optimization. Tech. Rep. DEIOC 3/2008, Dep. of Statistics, O.R. and Computing, University of La Laguna, Tenerife, Spain.
8. Geem, Z. W., Kim, J. H. and Yoon, Y.N. (2000), Optimal layout of pipe networks using harmony search, In: Proceedings of the 4th International Conference on Hydro-Science and Engineering, Seoul, South Korea.
9. Kashkooli, B.S and Monem, M.J., Simultaneous layout and pipe size optimization of pressurized irrigation water distribution networks using hybrid ACO/LIDM Method, Proceeding of the International Conference on Water Resources (ICWR 2009), Bayview Hotel, Langkawi, Kedah, Malaysia, May 26 – 27, 2009.
10. Kenedy, J, and Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of the International Conference on Neural Networks, Perth, Australia, 1995 IEEE, Piscataway, 1995. pp: 1942-1948.

11. Kennedy, J. and Eberhart, R. (1997), Adiscrete binary version of the particle swarm algorithm. In: IEEE Conference on Systems, Man, and Cybernerics, vol 5, pp 4104-4108.
12. Labye, Y., Iterative discontinuous method for networks with one or more flow regimes, In: Proceedings International Workshop on Systems Analysis of Problems in Irrigation, Drainage and Flood Control (pp 31–40). ICID, New Delhi, 1981.
13. Labye, Y., Olson, M. A., Galand, A. and Tsiourtis, N. (1988), Design and Optimization of Irrigation Distribution Networks, FAO Irrigation and Drainage Paper No. 44. FAO, Rome.
14. Lamaddalena, N. and Sagardoy, J. A. (2000), Performance analysis of on-demand pressurized irrigation systems, Irrigation and Drainage Paper no. 59. FAO, Roma.
15. Lejano, R. P. (2006), Optimizing the layout and design of branched pipeline water distribution systems, Irrigation and Drainage Systems (2006) 20: 125-137.
16. Montalvo, I., Izquierdo, J., P´erez, R. and Tung, M.M. (2008). Particle Swarm Optimization applied to the design of water supply systems. Computers and Mathematics with Applications, doi:10.1016/j.camwa.2008.02.006.
17. Moreno-Perez, J. A., Castro-Gutierrez, J. P, Martinez-Garcia F. J., Melian, B., Moreno-Vega, J. M. and Ramos, J. (2007), Discrete Particle Swarm Optimization for the p-median problem. In: Procceedings of the 7th Metaheuristics International Conference, Montreal, Canada.
18. Pereira, L.S., Calejo, M.J., Lamaddalena, N., Douieb, A. and Bounoua, R. (2003), Design and performance analysis of low pressure irrigation distribution systems, Irrigation and Drainage Systems, 17: 305–324.
19. Pugh, J., and Martinoli, A. (2006), Discrete multi-valued particle swarm optimization. In: Proceedings of IEEE Swarm Intelligence Symposium, vol 1, pp 103-110.
20. Reca, J., and Martinez, J. (2006), Genetic algorithems for the design of looped irrigation water distribution networks, Water Resour. Res., 42, W05416. Available on the WWW:url:http:// www.agu.org/pubs /crossref/2006/2005WR004383.shtml
21. Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C. and Wang, Q. X. (2007), Particle swarm optimization-based algorithms for TSP and generalized TSP, Information Processing Letters,103(2007) 169-176.
22. Trent, H. M. (1954), A note on the enumeration and listing of all possible trees in a connected linear graph. Proceedings, U. S. National Academy of Sciences, 40, (10), 1004-1007.
23. Walter, G.A. and Smith, D.K., Evolutionary Design Algorithm for Optimal Layout of Tree Networks, 1995, Engineering Optimization, 24:4,261 — 281
24. Yang, S. Wang, M. and Jiao, L. (2004), a Quantum Particle Swarm Optimization. In: Proceedings of CEC2004, the Congress on Evolutionary Computing, vol 1, pp 320-324.